Using GNG to improve 3D features extraction - Application to 6DoF Egomotion

Diego Viejo, Jose Garcia, Miguel Cazorla, David Gil

Instituto de Investigación en Informática
University of Alicante
PO. Box 99, E-03080 Alicante, Spain
Tel.: +34-965-903400 ext: 2072
Fax: +34-965-903902

Magnus Johnsson

Lund University Cognitive Science
Lundagard (Sweden)

Abstract

Several recent works deal with 3D data in mobile robotic problems, e.g. mapping or egomotion. Data comes from any kind of sensor such as stereo vision systems, time of flight cameras or 3D lasers, providing a huge amount of unorganized 3D data. In this paper, we describe an efficient method to build complete 3D models from a Growing Neural Gas (GNG). The GNG is applied to the 3D raw data and it reduces both the subjacent error and the number of points, keeping the topology of the 3D data. The GNG output is then used in a 3D features extraction method. We have performed a deep study in which we quantitatively show that the use of GNG improves the 3D feature extraction method. We also show that our method can be applied to any kind of 3D data. The 3D features obtained are used as input in an Iterative Closest Point (ICP)-like method to compute the 6DoF movement performed by a mobile robot. A comparison with standard ICP is performed, showing that the use of GNG improves the results. Final results of 3D mapping from the egomotion calculated are also shown.

Keywords: Egomotion, GNG, 3D feature extraction, 6DoF registration

1. Introduction

One of the main research topics in mobile robotics is the determination of the movement performed by the robot using its sensor information. The methods related to this research are called pose registration and can be used for automatic map building and SLAM Dissanayake et al. (2001). Our main goal is to perform six degrees of freedom (6DoF) pose registration in semi-structured environments, i.e., man-made indoor and outdoor environments. This registration can provide a good start point for SLAM. We use dense raw 3D data as input sets. Our method is developed for managing 3D point sets collected by any kind of sensor. For our experiments, we use three main data sources: a sweeping unit with a 2D laser Sick, a Digiclops stereo camera and an infrared camera SR4000, mounted on a mobile robot. Sweeping laser provides 3D data with a low error and a higher range compared to stereo systems, but data from this sensor is slower to retrieve than the other two systems. We are also interested in dealing with outliers, i.e., environments with people or non-modeled objects. This task is hard to overcome as classic algorithms, like ICP and its variants, are very sensitive to outliers. Furthermore, we don't use odometry information.

The extraction of feature points from 3D data has been extensively studied in the last years. Some examples of works that deal with this problem can be found in: Weingarten et al. (2004), Hähnel et al. (2003), Rusu et al. (2008), Steder et al. (2009), Steder et al. (2010), May et al. (2009) and Ruhnke et al. (2010).

This paper is focused on studying the movement performed by a mobile robot just using the environment information collected by its 3D sensor device. The trajectory followed by the robot can be reconstructed from the observations at each pose and thus a 3D map of the robot environment can be built. The problem of automatic map building has been challenging mobile robotics researchers during the last years Agrawal (2006); Koch & Teller (2007); Goecke et al. (2007).

Handling raw 3D data is not suitable for most of the mobile robot methodologies, due to the huge size of data that has to be managed. In our previous work Viejo et al. (2011) we proposed a method for extracting and modeling planar patches from 3D raw data. Using this method we achieve two main advantages: first, a complexity reduction (when comparing with raw data) is done and time and memory consumptions are improved (we obtain over 500 features from 100000 3D points); second, outliers are better overcome using these features, as points not sup-

Email addresses: dviejo@dccia.ua.es (Diego Viejo), jgarcia@dtic.ua.es (Jose Garcia), miguel.cazorla@ua.es (Miguel Cazorla), dgil@dtic.ua.es (David Gil), Magnus.Johnsson@lucs.lu.se (Magnus Johnsson)

ported by a planar patch are deleted. Planar patches are useful features as man-made environments are easily described with them. Our method can extract and model planar patches from 3D raw data, although it can be applied to other applications, like Villaverde & Granya (2009), Munyoz-Salinas et al. (2008) or Katz et al. (2010).

Nevertheless, in some situations, the planar patches extraction method is not sufficient to obtain a complete environment model. As we will explain later in this article, this kind of problems arise when the 3D sensor used combines both a short measurement range and a high measurement error. In these situations, we propose the use of a Growing Neural Gas that by means of competitive learning produces an adaptation of the reference vectors of the neurons as well as an interconnection network among them; thus obtaining a mapping that tries to preserve the topology of the input space. Besides, the network is capable of a continuous re-adaptation process even if new patterns are added, with no need to reset the learning. These features allow to represent fast and high quality 3D spaces, thus obtaining an induced Delaunay Triangulation of the input space very useful to obtain easily features like corners, edges, etc. We modify the original GNG method in order to be applied to data sequences: the GNG is adapted sequentially, i.e., the result in a given frame is taken as an input in the next frame. Modeling 3D scenes using GNG produces a more detailed result and thus further computations such as planar patches based egomotion are also improved. Some 3D reconstruction applications of Neural networks can be found in Cretu et al. (2008), do Rego et al. (2007), Ivrissimtzis et al. (2003), Fanany & Kumazawa (2004) and Holdstein & Fischer (2008). However, none of them manages 3D sequences of data.

In this paper, we show some new experiments, both quantitative and qualitative, proving the validity of our initial method. We also show that our method can be applied to any kind of 3D data. Besides, we describe deeply the main device used in this work, the Time-Of-Flight camera SR4000.

The rest of the paper is organized as follows: first, Section 2 describes the physical systems used for experiments; then, in Section 3 the GNG algorithm is explained and in Section 4 the algorithm for feature extraction is described; we continue with Section 5 where some results of 6DoF egomotion are shown; Finally, some conclusions are drawn and future work is commented in the last section.

2. Data acquisition

One of the goals of our work is to create an algorithm independent of the data source, i.e. to be applied to any robot platform, any 3D measurement device and can be used in indoor and outdoor man-made environments. We present here the physical systems used in our experiments.

We have used several robot platforms, depending on the perception system used. In Figure 1, two of these platforms are shown. The left one is an indoor platform, a Magellan Pro from iRobot. It is used for indoor experiments, given its dimensions (diameter: 40cm, height: 24cm). For outdoors we have used a PowerBot from ActiveMedia. It has a battery life of 5 hours, which is necessary for long outdoors experiments. Furthermore, PowerBot can carry the 3D sweeping laser unit, which is very heavy. Both come with an onboard computer.

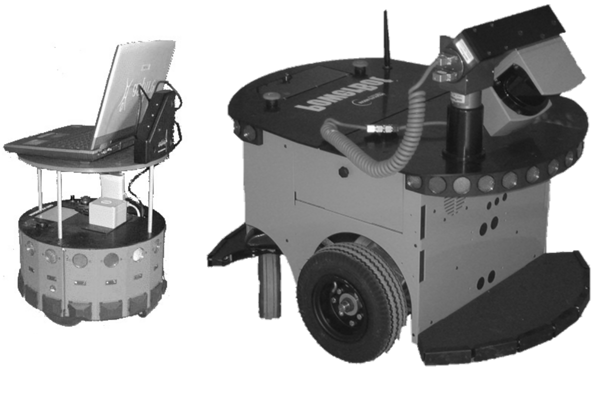

Figure 1: Mobile robots used for experiments. Left: Magellan Pro unit used for indoors. Right: PowerBot used for outdoors.

We use 3D data that come from different devices. First, we use a stereo camera Digiclops from Point Grey. It has a range of 8 meters and is ideal for indoor environments. It can provide 24 images per second with a grey level information for each point. However, it suffers the lack of texture: areas in the environment without texture can not provide 3D data. Furthermore, it has a measurement error of 10%. For outdoor environments we use a 3D sweeping laser unit, a LMS-200 Sick laser mounted on a sweeping unit. It does not suffer the lack of texture and its range is 80 meters with an error of 1mm per meter. The main disadvantage of this unit is the data capturing time: it takes more than one minute in one shot.

Finally, we have just acquired a SR4000 camera, which is a time-of-flight camera, based on infrared light. Time-of-Flight (ToF) cameras have been developed as a new technology that obtains range (distance) and amplitude maps by the use of a modulated light source. The main advantages with respect to other 3D devices are the possibility to acquire data at video frame rates and to obtain 3D point clouds without scanning and from just one point of view. The egomotion experiment of this paper is carried out using this camera, although other different devices were used, in order to show the validity of our method with different 3D sources. This was done because the SR4000 camera provides more outliers than the other devices used.

In our tests all the data was acquired directly from the camera, which obtains point coordinates XYZ, amplitude data of the scene and a confidence map of the distance measurements. In particular, the confidence map is obtained using a combination of distance and amplitude measurements and their temporal variations: it represents the probability that the measurement of the distance in each pixel is correct, so it can be useful to select regions containing high quality measurements or to reject low quality ones. In our experiments the amplitude data has low contrast so they have been equalized. Figure 2 shows an overview of the typical data obtained using a SR4000. The recorded 3D points cloud can be observed in the figure top center, corresponding amplitude on the left side and confidence on the right. Reference camera coordinates system is also shown.

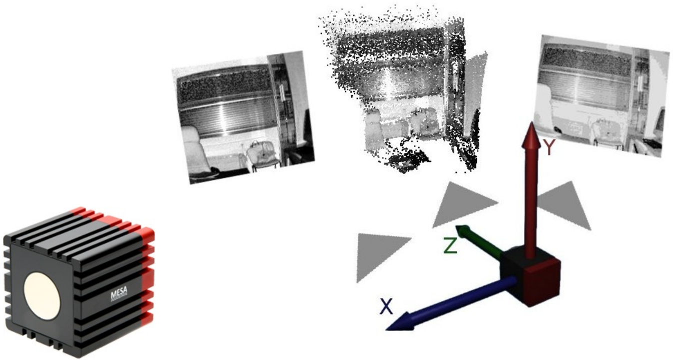

Figure 2: Left: SR4000 camera. Right: camera data overview. The SR4000 captures both a 3D point set and two maps: amplitude (left) and confidence (right).

ToF cameras allow the generation of point clouds during real time acquisition. The accuracy of ToF cameras varies depending on the internal components and the characteristics of the observed scene, such as objects reflectivity and ambient lighting conditions. These errors cannot be fully eliminated, but they can be reduced and optimized thanks to filtering or several techniques, such us averaging techniques or calibration procedures Chiabrando et al. (2009) where the proposed distance error model provided a reduction of distance errors in the 1.5-4m distance measurement range.

The basic principle of ToF cameras consists of an amplitude-modulated infrared light source and a sensor field that measures the intensity of backscattered infrared light. The infrared source is constantly emitting light that varies sinusoidally. Objects located at different distances are reached by different parts of the sinusoidal wave. The reflected light is then compared to the original one, calculating the phase shift, by means of measuring the intensity of the incoming light since the phase shift is proportional to the time of flight of the light reflected by a distant object. A detailed description of the time-of-flight principle can be found in Gokturk et al. (2004).

3. GNG Algorithm

Using Growing Neural Gas (GNG) Fritzke (1995) a growth process takes place from minimal network size and new units are inserted successively using a particular type of vector quantization Kohonen (2001). To determine where to insert new units, local error measures are gathered during the adaptation process and each new unit is inserted near the unit which has the highest accumulated error. At each adaptation step a connection between the winner and the second-nearest unit is created as dictated by the competitive Hebbian learning algorithm. This is continued until an ending condition is fulfilled, as for example evaluation of the optimal network topology or time deadline. In addition, in GNG networks learning parameters are constant in time, in contrast to other methods whose learning is based on decaying parameters. In the rest of this Section we describe the growing neural gas algorithm and the ending condition used in this work. The network is specified as:

- A set N of neurons (nodes). Each neuron $c \in N$ has its associated reference vector $w_c \in R^d$. The reference vectors can be regarded as positions in the input space of their corresponding neurons.
- A set of edges (connections) between pairs of neurons. These connections are not weighted and its purpose is to define the topological structure. An edge aging scheme is used to remove connections that are invalid due to the motion of the neuron during the adaptation process.

The GNG learning algorithm used to map the network to the input manifold is as follows:

1. Start with two neurons a and b at random positions w_a and w_b in R^d.
2. Generate at random an input pattern ξ according to the data distribution $P(\xi)$ of each input pattern.
3. Find the nearest neuron (winner neuron) s_1 and the second nearest s_2.
4. Increase the age of all the edges emanating from s_1.
5. Add the squared distance between the input signal and the winner neuron to a counter error of s_1 such as:
$$\triangle error(s_1) = \|w_{s_1} - \xi\|^2 \tag{1}$$
6. Move the winner neuron s_1 and its topological neighbors (neurons connected to s_1) towards ξ by fractions ϵ_w and ϵ_n, respectively, of the total distance:
$$\triangle w_{s_1} = \epsilon_w(\xi - w_{s_1}) \tag{2}$$
$$\triangle w_{s_n} = \epsilon_w(\xi - w_{s_n}) \tag{3}$$
for all direct neighbors n of s_1.
7. If s_1 and s_2 are connected by an edge, set the age of this edge to 0. If it does not exist, create it.
8. Remove the edges when age is larger than a_{max}. If this results in isolated neurons (without emanating edges), remove them as well.
9. Every certain number λ of input patterns generated, insert a new neuron as follows:
   - Determine the neuron q with the maximum accumulated error.
   - Insert a new neuron r between q and its further neighbor f:
   $$w_r = 0.5(w_q + w_f) \tag{4}$$
   - Insert new edges connecting the neuron r with neurons q and f, removing the old edge between q and f.
10. Decrease the error variables of neurons q and f multiplying them by a constant α. Initialize the error variable of r with the new value of the error variable of q and f.
11. Decrease all error variables by multiplying them by a constant γ.

12. If the stopping criterion is not yet achieved (in our case the stopping criterion is the number of neurons), go to step 2.

With regard to the processing of image sequences, we have introduced several improvements to the network in order to accelerate the representation and allow the architecture to work faster.

The main difference with the original GNG algorithm is the omission of insertion/deletion actions (steps 8 to 11) after the first frame. Since no neurons are added or deleted the system keeps correspondence during the whole sequence, solving intrinsically this problem. For the initial moment t_0 the representation is obtained making a complete adaptation of a GNG. However, for the following frames the previous network structure is employed. Thus, the new representation of the object is obtained by performing the iteration of the internal loop of the learning algorithm of the GNG, relocating the neurons and creating or removing edges. This adaptive method is also able to face real-time constraints, because the number λ of times that the internal loop is performed can be chosen according to the available time between two successive frames that depends on the acquisition rate.

For the experiments, the GNG parameters used are: $N = 2000$, $\lambda = 2000$, $\epsilon_w = 0.1$, $\epsilon_n = 0.001$, $\alpha = 0.5$, $\beta = 0.95$, $\alpha_{max} = 250$. In Figure 3 a result of applying GNG to a set of 3D points obtained from a SR4000 is shown, the same for stereo data in Figure 4 and, finally, in Figure 5 the resulting GNG for 3D laser data set can be observed.

In the case of a low rate of data acquisition, the representation obtained for a frame could be very different from the previous one and GNG should be reinitialized. Since GNG learned from the previous frame differs too much and should not be used as a started point for the next frame.

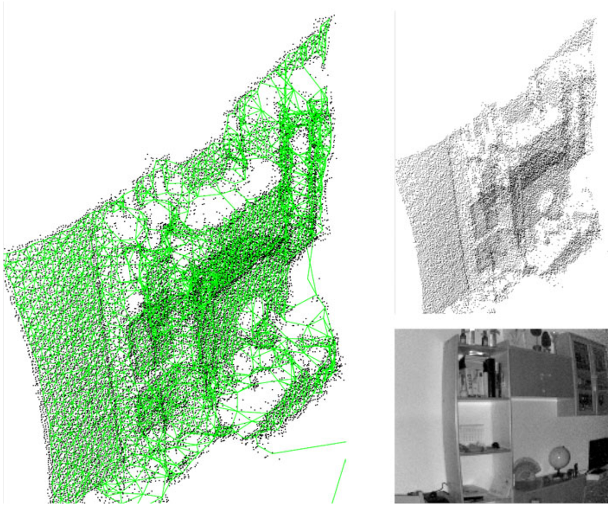

Figure 3: Applying GNG to SR4000 data set. The original data set can be observed at the top right corner. The intensity image captured by the camera is shown at the bottom right corner. Left, the resulting GNG (green) can be observed over the original data set.

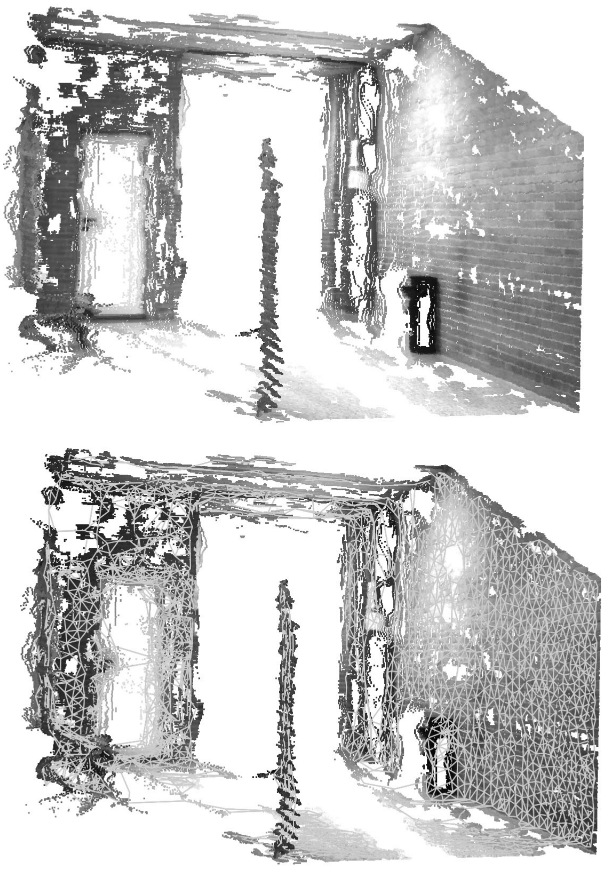

Figure 4: Applying GNG to stereo data set.

4. Features extraction method

We can still reduce more the amount of information contained in a 3D scene by modeling object surfaces included in it. Normal vectors estimated from a local area around each 3D point into the scene are a good starting point for obtaining surfaces descriptions. Some methods, such as Page et al. (2002) or Mitra & Nguyen (2003) were developed for handling noisy input data sets. The basic idea consists in analyzing each point in the local neighborhood by means of a robust estimator. In Martín et al. (2004) a singular value decomposition (SVD) based estimator is used to obtain surface normal vectors. Using this method, when the underlying surface is a plane, the minimum singular value is quite smaller than the other two singular values, and the singular vector related to the minimum singular value is the normal vector of the surface at this point.

From this information we can label each point in a 3D scene as belonging to a planar surface, when one of the singular values is much smaller than the others, or to not defined objects in other case. In Figure 6, we can see an example of the application of this segmentation for both outdoor and indoor scenes. Despite the segmentation of the scene points, we need to do some extra work to extract planar patches from the scene. We use a template matching to fit the labelled points into a planar patch model.

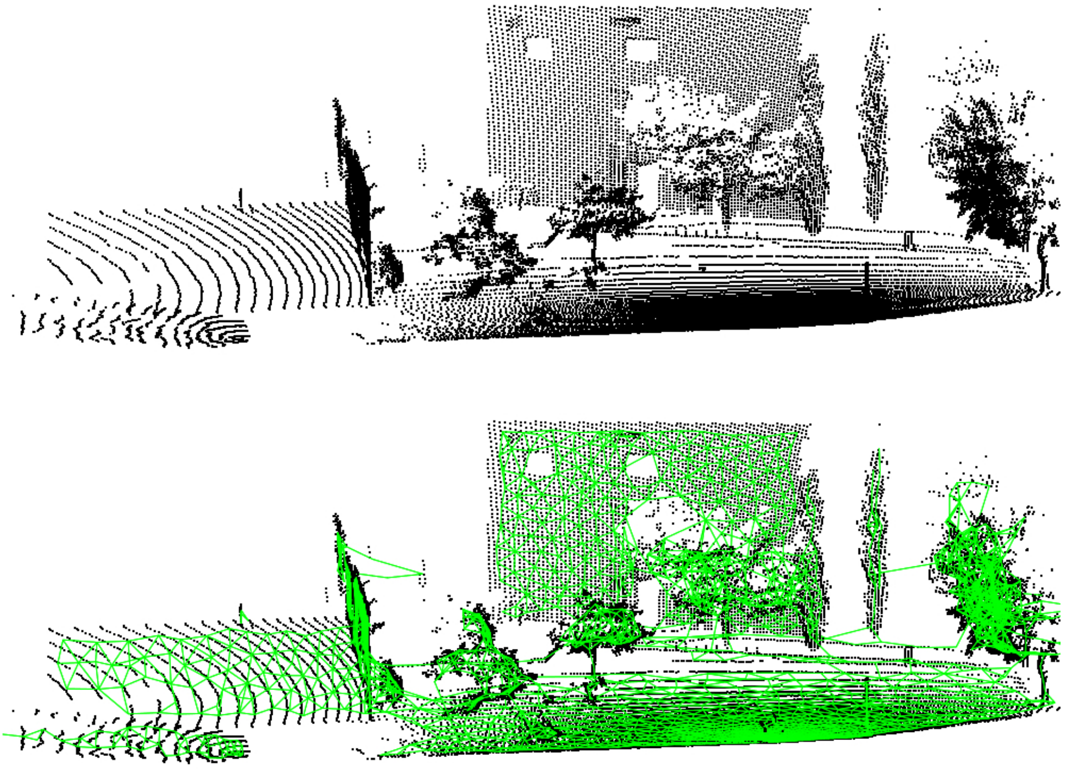

Figure 5: Applying GNG to a 3D laser set. Top: original 3D point set. Bottom: the GNG network (green) over the data set.

This process retrieves the underlying surface normal vector of a given set of points. Furthermore, a threshold called thickness can be defined from singular values in order to determine in which situations a point, as well as its neighborhood, belong to a planar surface or not. This thickness value can be used to measure how a 3D point set fits to a plane. The lower the thickness value we find, the better the fitting between the points and the planar surface is. The size of the window used to obtain neighbor points has an important impact on the results. As it is considered in Cole & Newman (2005), sample density of 3D laser range finder data presents large variations due to the divergence of consecutively sampled beams. In general, this characteristic is present in any 3D data set, independently of the sensor used. A complete study on the impact of different window size was performed in Viejo & Cazorla (2008). Summarizing, a depth-based adaptive window provides better results. Depending on the sensor measurement error, this window has to be set up at different starting sizes. The bigger the measurement error is, the bigger window size has to be set up.

Using SVD based normal vector estimation method we can obtain a model that represents the planar surfaces in the scene. We propose an optimal method that can obtain a planar patch model from a 3D point set in O(log n). This method is based on automatic seed selection methods Shih & Cheng (2005) Fan et al. (2005). The idea consists in performing a selection of the most representative points in the whole 3D scene. These selected points must belong to planar surfaces. To ensure this we use the thickness value, that can be obtained from the SVD based estimation method as we described above. In order to find out the most representative points, we randomly select points in the scene until all points are visited. For each point visited we compute its normal vector and thickness value. If its thickness value is low enough, the point is inserted into the most representative points list and its neighbors inside the window used to compute its normal vector are marked as visited. When this process ends, planar patches model is directly computed from the most representative points and its normal vectors. The size of the planar patches depends on the size of the window used to compute normal vectors. The overall procedure for fast planar patches estimation is as follow:

```
function Fast_Patch_Estimation ($\Pi$: 3DPoint_set; $thick : \mathbf{R}$)
  return: planar_patch_set
  var $p_i$: 3DPoint; $Q, neighs$: 3DPoint_set; $w : \mathbf{R}$;
      $< \mathbf{n_i}, \gamma_i >$: normal-and-thickness_tuple; $N : \mathbf{N}$
  begin
    Q := $\Pi$
    while ($Q \neq \emptyset$) do
      $N := \|Q\|$
      $p_i :=$ Remove($Q$, Random($N$))
      $w :=$ ComputeWindowSize($p_i$)
      $neighs :=$ GetNeighbours($\Pi, p_i, w$)
      $< \mathbf{n_i}, \gamma_i > :=$ NormalSVD($p_i, neighs$)
      if ($\gamma_i < thick$) then
        Add($result$, newPlanarPatch($p_i, \mathbf{n_i}, w$))
```

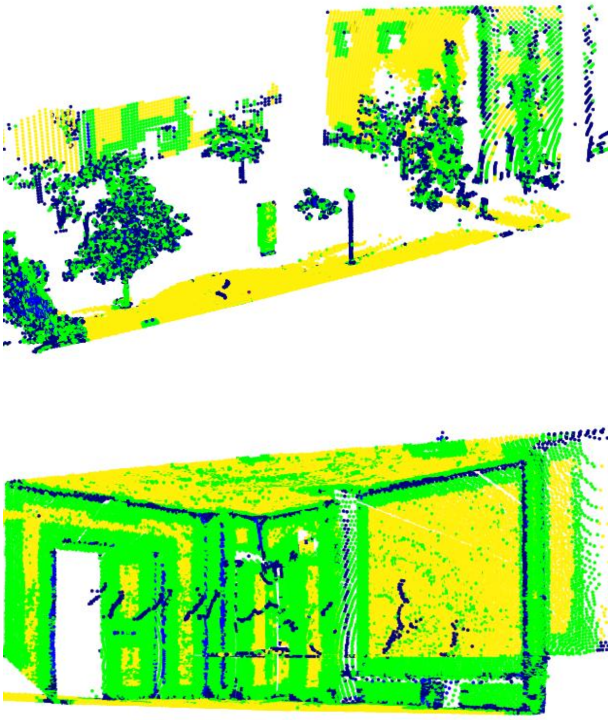

Figure 6: The points labelled in yellow color represent planar surfaces. The points labelled in green and blue represent another kind of surfaces.

```
            RemoveAll($Q, neighs$)
        **endif**
    **endwhile**
    Fast_Patch_Estimation := $result$;
**end**.
```

In Figure 7 we can observe the result of applying this method for computing planar patches models from 3D scenes captured by a 3D range laser. Since our method is supposed to work using any data source, the result of applying it to a stereo 3D image can be observed in Figure 8.

As we stated before, the window size (Parameter w in function *GetNeighbours*) is a key factor. It depends on both the depth of the point and the 3D sensor measurement error. Bigger windows will produce better results for noisy 3D sets but will also discard small objects, compared to window size in the scene. This lack of small details may lead to problems in further computations, specially when 3D sensors with a short range (up to 10 meters) are used. Frequently, under this configurations the sensor can not obtain important information such as the end of a corridor or the bottom part of a big room and these small details are really important. To overcome this problem in this paper we introduce the use of GNG in order to improve the feature extraction method. GNG produces a Delaunay Triangulation which can be used as a representation of the points neighborhood. In this way we can state the neighbor searching according to GNG and produce more detailed and accurate planar patches descriptions. Figure 9 shows planar patches extraction from a 3D image obtained by a SR4000 camera. The bottom image shows the results of combining GNG with the features extraction procedure. It can be compared with the upper image in which no GNG has been used.

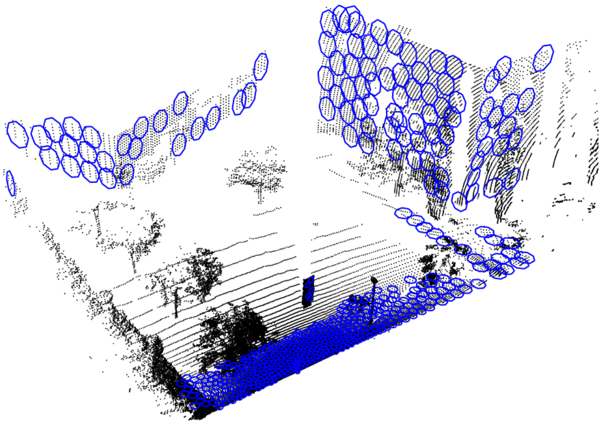

Figure 7: Planar patches extracted from 3D laser range finder data. Patches are represented by blue circles. Radius of each patch depends on the size of the window used to compute the patch.

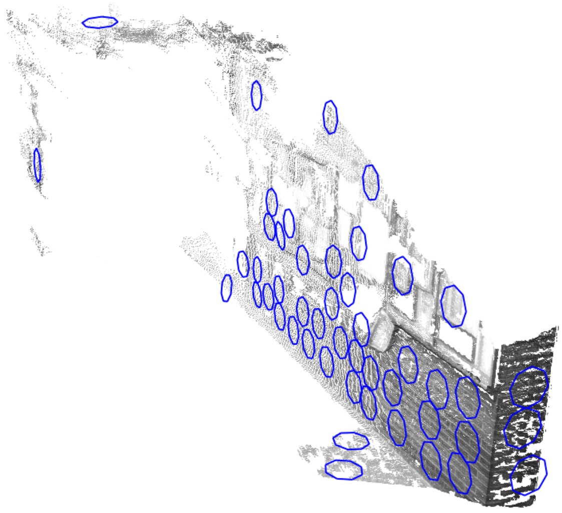

Figure 8: Planar patches extracted from a stereo scene.

5. Using 3D models: 6DoF egomotion

In the previous section we described a method for building 3D models from scenes captured by a 3D sensor. Furthermore, we want to use these models to achieve further mobile robot applications in real 3D environments. The basic idea is to take advantage of the extra knowledge that can be found in 3D models such as surfaces and its orientations. This information is introduced in a modified version of an ICP-like algorithm in order to

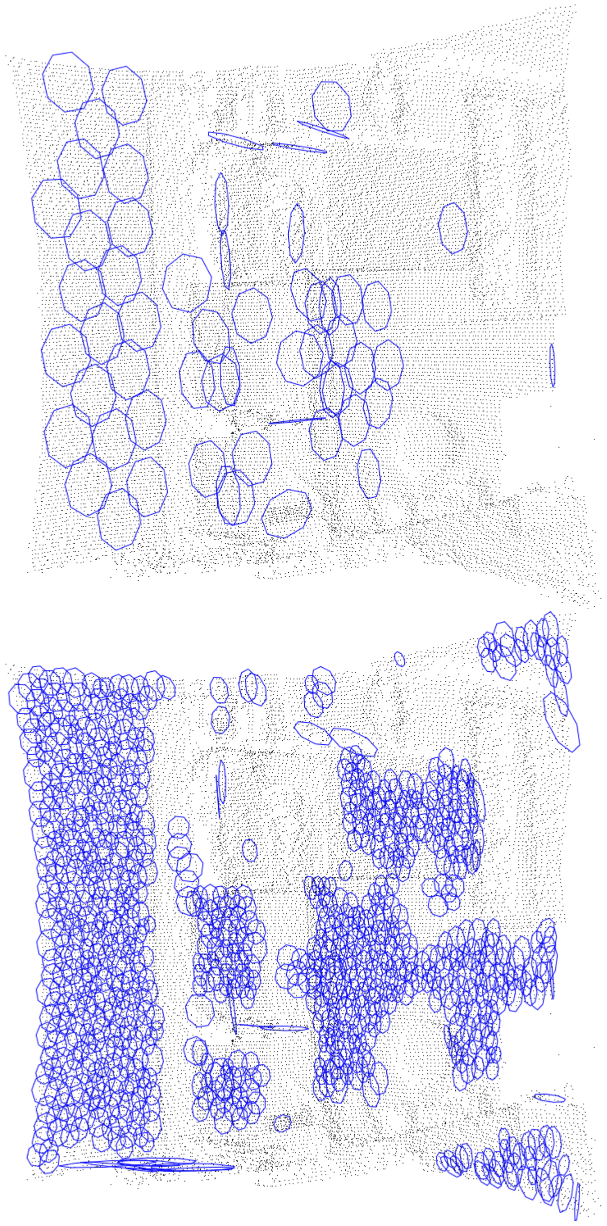

Figure 9: Planar patches extracted from SR4000 camera. The image in the bottom uses GNG for improving planar patches extraction. As a result we obtain a more detailed planar patches descriptions

reduce the outliers incidence in the results. ICP Besl & McKay (1992); Zhang (1994) is widely used for geometric alignment of a pair of three-dimensional points sets. From an initial approximate transformation, ICP iterates the next three steps until convergence is achieved: first, closest points between sets are stated; then, best fitting transformation is computed from corresponding points; finally, transformation is applied. In the mobile robotics area, the initial transformation usually comes from odometry data.

Nevertheless, our approach does not need an initial approximate transformation like ICP based methods do. We can use the global model structure to recover the correct transformation. This feature is useful for situations where no odometry is available or it is not accurate enough, such as legged robots. In our case, we are going to exploit both the information given by the normal vector of the planar patches and its geometric position. Whereas original ICP computes both orientation and position at each iteration of the algorithm, we use the advantage of knowing planar patches orientation to decouple the computation of rotation and translation. Thus, we first register the orientation of planar patch sets and when the two planar patches sets are aligned we address the translation registration. In Figure 10 we can observe the steps performed for computing the alignment between two sets of planar patches. The top image shows a zenithal view of two planar patches sets computed from two consecutive 3D scenes obtained by a robot during its trajectory. The middle image shows the result of rotation registration. Finally, bottom image shows the result after the translation between planar patches sets is computed.

In the next experiment we compare quantitatively the performance of our algorithm with the classical ICP one. To establish a ground truth, we put the SR4000 camera on a rotating PowerCube unit. This unit allows precise rotations around the Y axis. We take 3D images with the camera every 2 degrees (0.035 radians) until a full rotation is completed. The camera is not affected by any translation movement, except rotation. With the information provided by the PowerCube unit we can set the alignment error that occurs when each pair of images are matched. Figure 11 (top) shows the translation error for the alignment of consecutive images from the experiment. In the same way, Figure 11 (bottom) shows the angular error committed. In both cases, our proposal is compared with the results of vanilla ICP algorithm. Both in position and orientation, most of the time the error committed by our proposal is smaller than the error committed by the ICP algorithm. In absolute terms, the root mean square error (RMS) for the alignment of the position using ICP algorithm is 0.072m. while our method reduces it to 0.039m. As for the orientation, the ICP algorithm RMS error is 0.030 rad, whereas our approach reduces this error to 0.019 rad. This experiment demonstrates that our approach outperforms ICP results. The next examples show our results of automatic map building in real scenarios. We will also compare the resulting map obtained using our approach with the one obtained by the ICP algorithm.

In Figures 12, 13 and 14 we show an example of 3D map building using this 6DoF egomotion approach. For this experiment, 100 3D images from a 5 meter range SR4000 camera have been used. The first figure shows a 3D view of the reconstructed environment using ICP for 6DoF egomotion reconstruction. In Figure 13 the same scene is reconstructed by using the planar patches based egomotion, but without using GNG. In this figure we can observe that the result is improved, but it still does not provide good results. Finally, Figure 14 shows the result using GNG for improving feature extraction. The better the feature extraction process is, the better reconstruction is obtained. While in the first two experiments the registration of the sequence was almost impossible, in the last one the reconstruction was reasonably good.

The ICP algorithm takes an average time of 40s. to find a registration between two consecutive poses. The first computa-

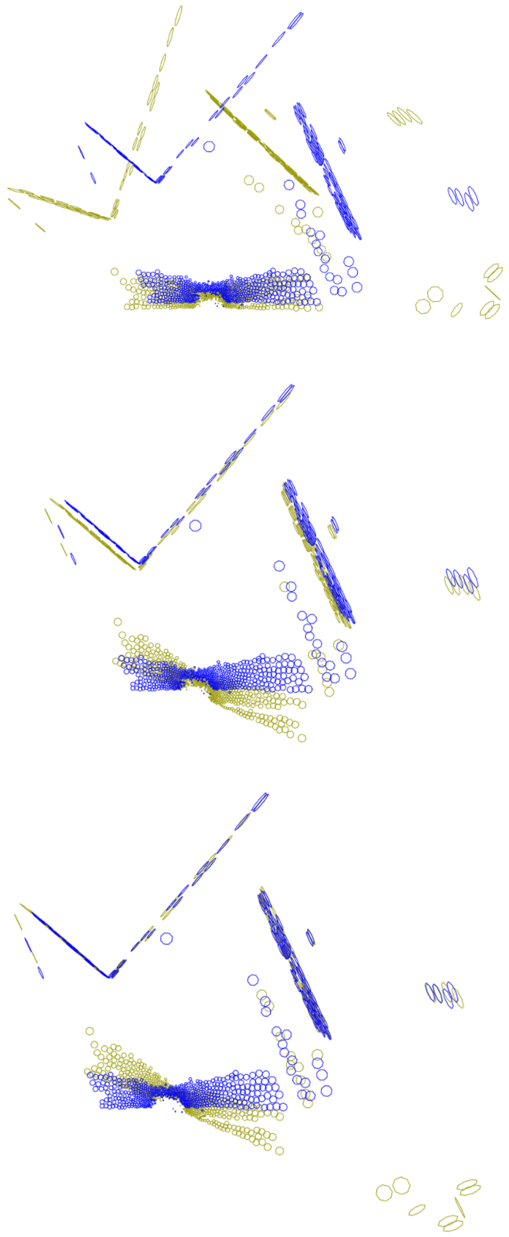

Figure 10: Planar patches matching example. For all the three images patches from the model are painted in dark grey whereas scene paths are represented in light grey. Top, initial situation. Middle, after rotation registration. Bottom, final result after translation registration is completed.

tion using the GNG takes almost 100s. To that, we need to add a mean time of 4s. for calculating the planar patches. And 100ms for finding the registration between poses. However, using our adaptive method, in the subsequent poses the GNG adaptation takes only 1s. Thus, for the first two poses, our method takes 109,2s., but from then on it takes only 10,2s., compared with the 40s. of the ICP.

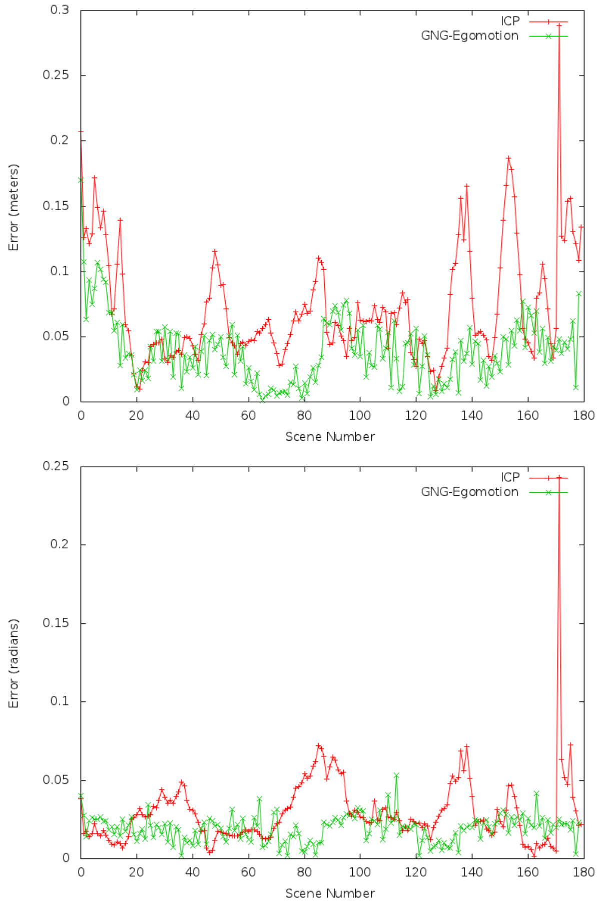

Figure 11: ICP versus Egomotion-GNG error comparison. Top, position error. Bottom, angular error.

6. Conclusions and future work

We have presented a new method for computing 3D models of semi-structured environments from unorganized raw 3D data. Since our method works with raw 3D point sets, we do not need to know anything about the kind of sensor used for obtaining data. The proposed method can be used with most of the 3D scanner devices. First, we have explained an algorithm for computing the planar patches that fits with the planar surfaces in the 3D scene. This is a low complexity method that can be used to obtain online 3D models. We have also used a GNG model in order to increase the level of details for the resulting models. Results have been shown for both time-of-flight cameras and 3D range laser. The usefulness of our models is demonstrated when applying a 6DoF egomotion algorithm that uses these models as an input for computations. It has been proven that the use of GNG improves 6DoF mapping results, thus making it possible to use this approach with any kind of 3D sensor.

The proposed modification of GNG to represent 3D data sequences accelerates the learning algorithm and allows the architecture to work faster. This is possible because the system does not restart the map for each frame in the sequence, it only readjusts the network structure starting from previous map without

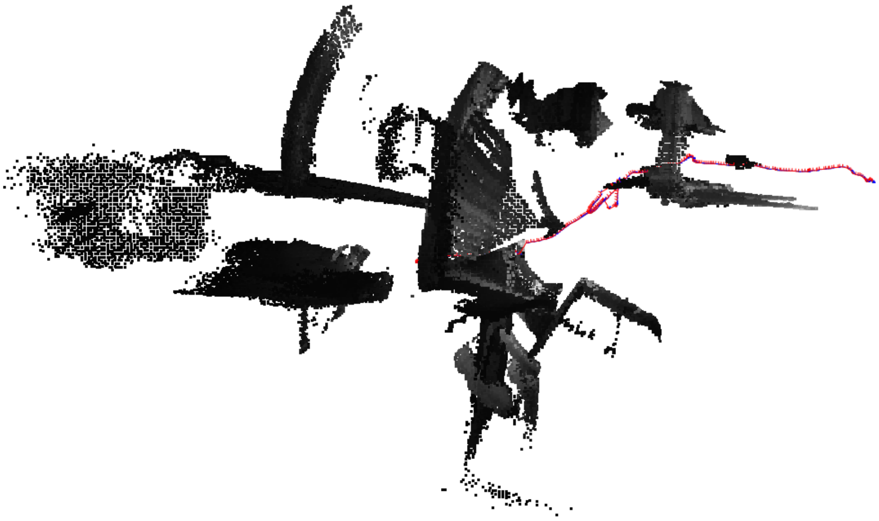

Figure 12: 6DoF egomotion results using ICP.

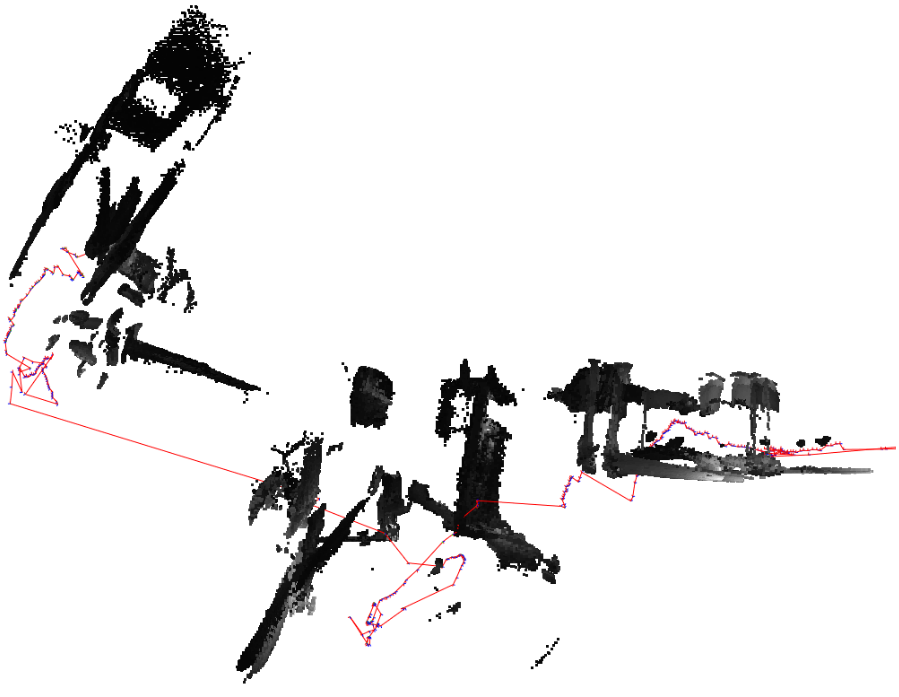

Figure 13: Planar based 6DoF egomotion results using planar patches method without using GNG.

inserting or deleting neurons.

The experiments show that the proposed method can be applied to the problem of automatic map reconstruction. The results show significant improvements over the classical solutions. The use of GNG to improve the feature extraction process has been a key aspect in improving 6DoF egomotion computa-

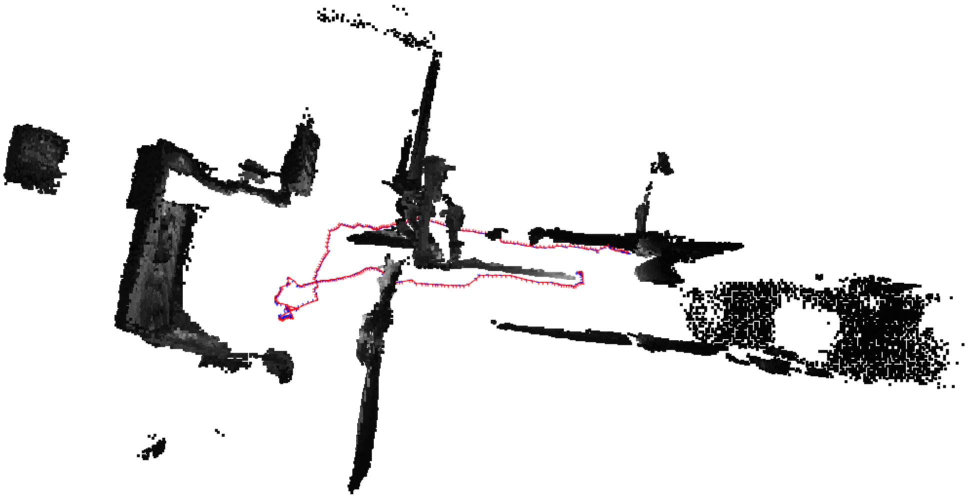

Figure 14: Planar based 6DoF egomotion results using planar patches after computing a GNG mesh.

tion.

As future work we plan to improve the accuracy and performance of our method in order to use it in 6D SLAM. We also plan to study how merging different planar patches could increase the accuracy and/or time registration.

Acknowledgments

This work has been partially supported by grant DPI2009-07144 from Ministerio de Ciencia e Innovacion of the Spanish Government and by the University of Alicante projects GRE09-16 and GRE10-35, and Valencia's Government project GV/2011/034.

References

Agrawal, M. (2006). A lie algebraic approach for consistent pose registration for general euclidean motion. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1891–1897).

Besl, P., & McKay, N. (1992). A method for registration of 3-d shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *14*, 239–256.

Chiabrando, F., Chiabrando, R., Piatti, D., & Rinaudo, F. (2009). Sensors for 3d imaging: Metric evaluation and calibration of a ccd/cmos time-of-flight camera. *Sensors*, *9*, 10080–10096.

Cole, A. R. H., David M., & Newman, P. M. (2005). Using naturally salient regions for slam with 3d laser data. *Proc. of the IEEE International Conference on Robotics and Automation*, .

Cretu, A.-M., Petriu, E., & Payeur, P. (2008). Evaluation of growing neural gas networks for selective 3d scanning. In *Robotic and Sensors Environments, 2008. ROSE 2008. International Workshop on* (pp. 108 –113).

Dissanayake, M., Dissanayake, M., Newman, P., Clark, S., Durrant-Whyte, H., & Csorba, M. (2001). A solution to the simultaneous localization and map building (slam) problem. *Robotics and Automation, IEEE Transactions on*, *17*, 229–241.

Fan, J., Zeng, M., Body, M., & Hacid, M. S. (2005). Seeded region growing: an extensive and comparative study. *Pattern Recognition Letters*, *26*, 1139–1156.

Fanany, M. I., & Kumazawa, I. (2004). A neural network for recovering 3d shape from erroneous and few depth maps of shaded images. *Pattern Recogn. Lett.*, *25*, 377–389.

Fritzke, B. (1995). A growing neural gas network learns topologies. In *Advances in Neural Information Processing Systems* (pp. 625–632). MIT Press volume 7.

Goecke, R., Asthana, A., Pettersson, N., & Petersson, L. (2007). Visual vehicle egomotion estimation using the fourier-mellin transform. In *Proc. IEEE Intelligent Vehicles Symposium* (pp. 450–455).

Gokturk, S. B., Yalcin, H., & Bamji, C. (2004). A time-of-flight depth sensor - system description, issues and solutions. In *CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 3* (p. 35). Washington, DC, USA: IEEE Computer Society.

Hähnel, D., Burgard, W., & Thrun, S. (2003). Learning compact 3d models of indoor and outdoor environments with a mobile robot. *Robotics and Autonomous Systems*, *44*, 15–27.

Holdstein, Y., & Fischer, A. (2008). Three-dimensional surface reconstruction using meshing growing neural gas (mgng). *Vis. Comput.*, *24*, 295–302.

Ivrissimtzis, I., Jeong, W.-K., & Seidel, H.-P. (2003). Using growing cell structures for surface reconstruction. In *Shape Modeling International, 2003* (pp. 78 – 86).

Katz, R., Nieto, J., & Nebot, E. (2010). Unsupervised classification of dynamic obstacles in urban environments. *J. Field Robot.*, *27*, 450–472.

Koch, O., & Teller, S. (2007). Wide-area egomotion estimation from known 3d structure. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR '07* (pp. 1–8).

Kohonen, T. (2001). *Self-Organising Maps*. Springer-Verlag.

Martín, M., Gómez, J., & Zalama, E. (2004). Obtaining 3d models of indoor environments with a mobile robot by estimating local surface directions. *Robotics and Autonomous Systems*, *48*, 131–143.

May, S., Droeschel, D., Holz, D., Fuchs, S., Malis, E., Nüchter, A., & Hertzberg, J. (2009). Three-dimensional mapping with time-of-flight cameras. *J. Field Robotics*, *26*, 934–965.

Mitra, N. J., & Nguyen, A. (2003). Estimating surface normals in noisy point cloud data. In *SCG '03: Proceedings of the nineteenth annual symposium on Computational geometry* (pp. 322–328). New York, NY, USA: ACM Press.

Munyoz-Salinas, R., Aguirre, E., Garcia-Silvente, M., & Gonzalez, A. (2008). A multiple object tracking approach that combines colour and depth information using a confidence measure. *Pattern Recognition Letters*, *29*, 1504 – 1514.

Page, D. L., Sun, Y., Koschan, A. F., Paik, J., & Abidi, M. A. (2002). Normal vector voting: crease detection and curvature estimation on large, noisy meshes. *Graph. Models*, *64*, 199–229.

do Rego, R., Araujo, A., & de Lima Neto, F. (2007). Growing self-organizing maps for surface reconstruction from unstructured point clouds. In *Neural Networks, 2007. IJCNN 2007. International Joint Conference on* (pp. 1900 –1905).

Ruhnke, M., Steder, B., Grisetti, G., & Burgard, W. (2010). Unsupervised learning of compact 3d models based on the detection of recurrent structures. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2137–2142).

Rusu, R. B., Marton, Z. C., Blodow, N., & Beetz, M. (2008). Learning Informative Point Classes for the Acquisition of Object Model Maps. In *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV), Hanoi, Vietnam, December 17-20*.

Shih, F. Y., & Cheng, S. (2005). Automatic seeded region growing for color image segmentation. *Image and Vision Computing*, *23*, 877–886.

Steder, B., Grisetti, G., & Burgard, W. (2010). Robust place recognition for 3D range data based on point features. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*.

Steder, B., Grisetti, G., Van Loock, M., & Burgard, W. (2009). Robust online model-based object detection from range images. In *Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems* IROS'09 (pp. 4739–4744). Piscataway, NJ, USA: IEEE Press.

Viejo, D., & Cazorla, M. (2008). 3d model based map building. In *International Symposium on Robotics, ISR 2008*.

Viejo, D., Garcia, J., Cazorla, M., Gil, D., & Johnsson, M. (2011). Using 3d gng-based reconstruction for 6dof egomotion. In *Proceedings of International Joint Conference on Neural Networks, IJCNN 2011*.

Villaverde, I., & Granya, M. (2009). An improved evolutionary approach for egomotion estimation with a 3d tof camera. In J. Mira, J. Ferrández, J. Álvarez, F. de la Paz, & F. Toledo (Eds.), *Bioinspired Applications in Artificial and Natural Computation* (pp. 390–398). Springer Berlin / Heidelberg volume 5602 of *Lecture Notes in Computer Science*.

Weingarten, J. W., Gruener, G., & Siegwart, R. (2004). Probabilistic plane fitting in 3D and an application to robotic mapping. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on* (pp. 927–932 Vol.1). volume 1.

Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, *13*, 119–152.